RFC 105

NIC 5775

Network Specifications For

Remote Job Entry

and

Remote Job Output Retrieval

at UCSB

22 March 71

James E. White
Computer Research Lab.
University of California
Santa Barbara, Calif.

In the discussions that follow, 'byte' means 8 bits, with those eight bits numbered 0-7 from left to right.

I - Remote Job Entry (RJE)

UCSB will accept input of pseudo card files for batch processing at socket number x'200', site 3. Network users should obtain an account number from the UCSB Computer Center; account #1025, programmer names 'UCLA', 'SRI', 'UTAH', etc. may be used during checkout. The 360/75 runs under OS MVT and HASP. Users submit jobs to HASP for scheduling and subsequent execution by OS through an intermediary process hereafter called RJE which is addressed as socket number x'200' and can be invoked through the Logger. This section is intended to provide programmers with the information necessary to communicate with RJE; the reader is assumed familiar with the batch services offered by the Computer Center, and with its job control language (JCL) requirements.

RJE conducts all Network transactions through the NCP, which operates under the Host-Host protocol of 3 August 70. It expects the first message it receives to be of Type 0, discards the first eight bits (the message type) assuming them to be zeros, and thereafter for the life of the connection takes no notice of IMP-message boundaries.

I.A - Logging into RJE

To submit one or more jobs for batch processing, the Network user must establish a simplex connection with RJE. RJE is core resident only while such a simplex connection is established (i.e., while a user is transmitting a file). At all other times, it resides on direct-access storage and must be invoked through the Logger. A login sequence can always be initiated by requesting connection to socket x'200'. RJE does not serve multiple users simultaneously. Thus if a connection request is made to that socket while RJE is in use, the NCP will queue the request. When the current file transmission is complete, RJE will listen for and accept the next request (if any) in its queue; if no requests are queued for it, it will terminate execution, releasing the main storage it occupied. At times when RJE is not in core, the Logger listens on socket x'200', and will reject the first call it receives, read RJE into core, and dispatch it. RJE will then listen on that socket. Thus to initiate a login sequence, the user requests connection to socket x'200'. If accepted, he is in contact with RJE. If rejected, he should reissue the connection request; when accepted, he will be connected to RJE. A second rejection would indicate that the NCP's resources were exhausted. Once the connection has been established, RJE will consider the user logged in.

To prevent RJE from being monopolized by a single user, provision is made within the software for terminating a connection if RJE is ever

required to wait more than a certain amount of time for a transmission from the connected user. For now, this time limit has been set at one minute per record, but it may be shortened or lengthened as required in the future. Barring such termination, RJE will maintain its connection to the user indefinitely. Card images will be accepted over the connection, and each one will be passed to HASP as it is received. The user is expected to close the connection once his file has been transmitted. RJE will interpret that action as an end-of-file indication, and the user will be considered logged off.

I.B - The RJE Connection

RJE expects the first byte of data it receives over the connection established with it to be zeros, indicating message Type 0; it discards this byte unexamined, and thereafter, attaches no significance to IMP-message boundaries. The second byte of data received is interpreted as flags specifying the format of the data (file) to follow. The byte is interpreted as follows:

```
Bits 0-1 = 00:  file follows as Class A (stream-oriented)
                input.
         = 01:  not defined, should not occur.
         = 10:  file follows as Class B (variable-length
                record) input.
         = 11:  file follows as Class C (fixed-length record)
                input.
Bits 2-7     :  not examined, should be zeros.
```

Once made, this declaration prevails for the life of the connection.

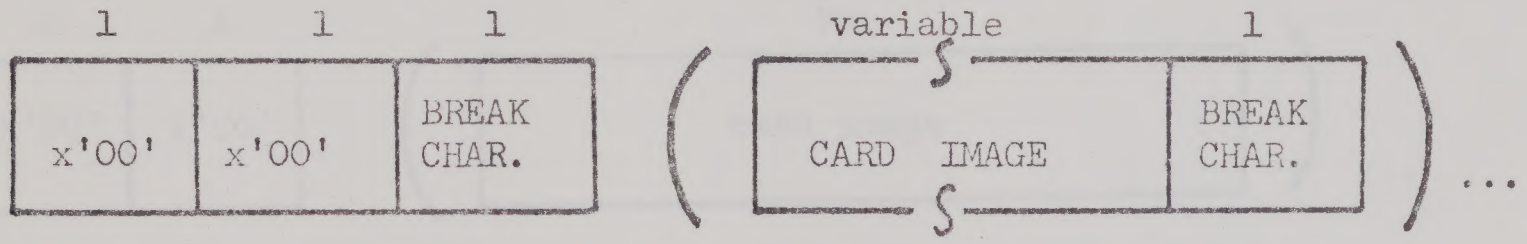
Regardless of the input class specified, the user transmits his file as card images, each of which will be padded on the right with blanks or truncated on the right to 80 bytes if necessary. The file transmitted must be structured exactly as if it were being placed on the card reader at the Computer Center. A job card and all the other, usual JCL must be present for each job in the file (batching of jobs is permissible and is transparent to RJE). For any job which requires that special (non-resident) disk(s) and/or tape(s) be mounted, a special JCL card must be inserted immediately after the job card for that job, and it must have the format:

/*SETUP vol-ser$_1$, vol-ser$_2$,...

where 'vol-ser$_i$' is the volume serial number of a volume requiring mounting. '/*SETUP' begins in column 1; 'vol-ser$_1$' must begin in column 16. The job will then enter the System in a HASP hold status until the required volume(s) can be mounted by the operator. If the user neglects to declare all such required volumes, his job is subject to immediate cancellation. All cards of the file which are not contained in a SYSIN data set must consist of valid, EBCDIC characters.

I.B.1 - Class A (Stream-Oriented) Input

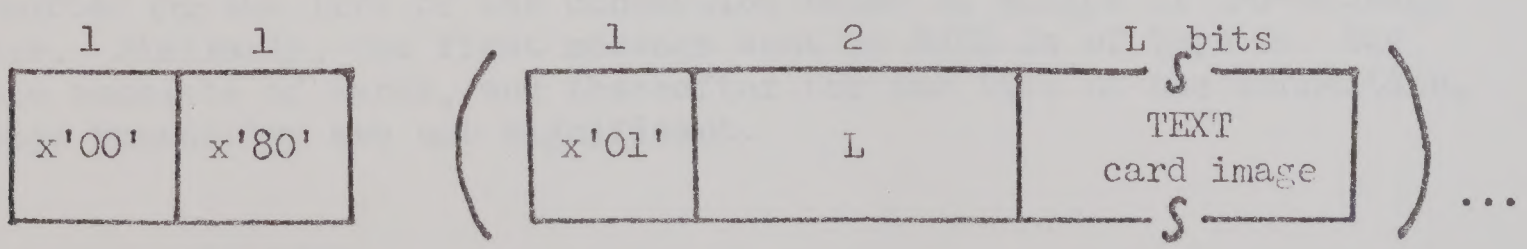
If input to RJE has been declared as Class A, the third byte of data received over the connection by RJE is interpreted as a break character declaration. Each byte received thereafter is compared to that character. Any other character is taken to be the next byte of the current card image. Whenever the break character is encountered, the previous byte is taken to be the last byte of the current card image, which is then padded or truncated as required and passed to HASP. Zero or more non-break characters may occur between occurrences of the break character. Thus when Class A input is specified, data transmitted to RJE shall have the following form:



where the length of each field has been specified in bytes. Zero or more occurrences of the quantity in parentheses may be transmitted before the connection is closed by the user.

I.B.2 - Class B (Variable-Length Record) Input

If input to RJE has been declared as Class B, then all input after the initial two bytes is expected to consist of a contiguous string of variable length records, each consisting of a one-byte op code (the op code should be x'01'), a two-byte length field which specifies the unsigned length in bits of the variable-length text field which follows. The text field may be zero or more bytes in length; the length field must contain an integer which is a multiple of 8. The text field represents one card image, which is padded or truncated by RJE as required and passed to HASP. Thus when Class B input is specified, data transmitted to RJE shall have the form:

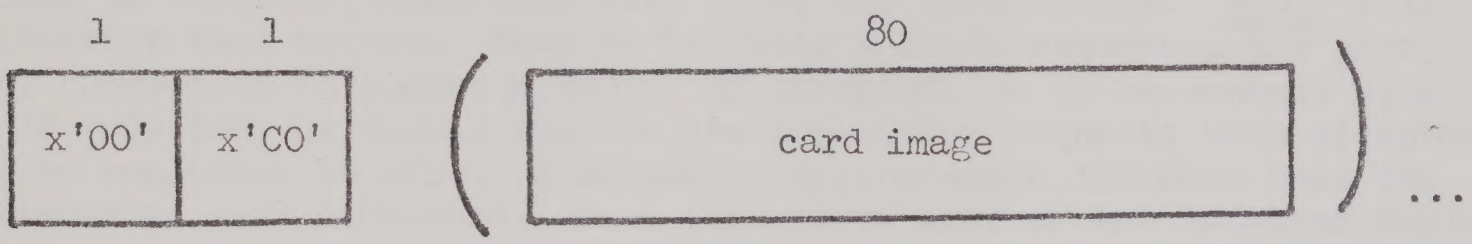


where the length of each field has been specified in bytes, except where

stated to the contrary. Zero or more occurrences of the quantity in parentheses may be transmitted before the connection is closed.

I.B.3 - Class C (Fixed-Length Record) Input

If input to RJE has been declared as Class C, then all input after the initial two bytes is expected to consist of a contiguous string of fixed-length, 80-byte card images. Thus, when Class C input is specified, data transmitted to RJE shall have the form:



where the length of each field has been specified in bytes. Zero or more occurrences of the quantity in parentheses may be transmitted before the connection is closed.

II - Remote Job Output Retrieval (RJOR)

Class A SYSOUT output from jobs submitted through RJE for batch processing at UCSB may be obtained by contacting socket x'300', site 3, provided that when the job was submitted, the character 'T' appeared as the eighth positional accounting parameter on the job card. Output is retrieved upon request and relayed to the Network user by a process hereafter called RJOR which is addressed as socket x'300'. RJOR can be invoked through the Logger. This section is intended to provide programmers with the information necessary to communicate with RJOR.

RJOR conducts all Network transactions through the NCP, which operates under the Host-Host protocol of 3 August 70. RJOR expects the first message it receives to be Type 0, discards the first byte, assuming it to be zeros, and thereafter for the life of the connection takes no notice of IMP-message boundaries. Similarly, the first message sent by RJOR is of Type 0: the first byte consists of zeros, and thereafter for the life of the connection, IMP-message boundaries are not significant.

II.A - Logging into RJOR

To obtain output from a batch-mode job, the Network user must establish

a full duplex connection with RJOR. RJOR is core resident only while in use (i.e., while control information or a file is being transmitted to or from a user, or while RJOR is waiting for a previously requested output file (or files)). At all other times, it resides on direct-access storage and must be invoked through the Logger. A login sequence can always be initiated by requesting connection to socket x'300'. If a connection request is made to that socket while another user is being logged in, the NCP will queue the request. After the current connection is terminated, RJOR will listen for and accept the next request (if any) in its queue; if no requests are queued for it and if it has fulfilled all of its output file requests, it will terminate execution, releasing the main storage it occupied. At times when RJOR is not in core, the Logger listens on socket x'300', and will reject the first call it receives, read RJOR into core, and dispatch it. RJOR will then listen on that socket. Thus to initiate a login sequence, the user requests connection to socket x'300'. If accepted, he is in contact with RJOR. If rejected, he should reissue the connection request; when accepted, he will be connected to RJOR. A second rejection would indicate that the NCP's resources were exhausted. Once this first half of the required duplex connection has been established, RJOR will consider the user logged in.

Over this first connection (hereafter called the Input Connection), the user transmits flags designating the function(s) to be performed by RJOR, and the name of the job to which the function(s) is(are) to be applied. RJOR then closes this connection. RJOR transmits control information specifying the disposition of the user's request and the output file (if requested) over a second connection involving RJOR's socket number x'301', site 3, and the socket at the user's site whose socket number is one less than that on which RJOR was contacted by the user. The user's request may or may not be immediately executable. If the former is the case, RJOR issues a connection request to the designated user receive socket, and when the connection is established transmits whatever control information is appropriate along with the user's output (if requested); RJOR then closes the connection and the user is considered logged out. If the user's request cannot be immediately satisfied (e.g., the job whose output is sought hasn't been submitted yet or hasn't finished execution), the second connection is opened by RJOR just long enough to inform the user of the delay, and then closed. Then when the request can be serviced, the connection is reopened, the required data transmitted, and the connection closed; the user is then considered logged out.

To prevent RJOR from being monopolized by a single user, provision is made within the software for terminating a connection if RJOR is ever required to wait more than a certain amount of time for completion of a transmission to or from the connected user. For now, this time limit has been set at one minute per record, but it may be shortened or lengthened as required in the future.

II.B - The Input Connection

RJOR expects the first byte of data it receives over the Input Connection to be zeros, indicating message Type 0; it discards this byte unexamined, and

thereafter, attaches no significance to IMP-message boundaries. The second byte of data received is interpreted as flags specifying the function(s) to be performed. Following the flag byte, RJOR expects an eight-byte, EBCDIC job name, padded on the right with blanks if necessary. The flag byte is interpreted as follows:

Bit 0 = 1: transmit the output generated by the specified job.
Bit 1 = 1: purge the output file created by the specified job.
Bit 2 = 1: wait as long as is required to execute the function(s) specified by Bits 0-1.
= 0: if the function(s) specified by Bits 0-1 cannot be executed immediately, simply reutrn an indication of that fact.
Bit 3 = 1: an earlier request pertaining to the specified job which exercised the wait-for-output (Bit 2) option is to be canceled.
Bits 4-7: not examined, should be zeros.

Any combination of Bits 0-2 is permissible. If Bit 3 = 1, no other bits are examined. If Bit 0=1 and Bit 1 = 1, the output file is transmitted before it is purged. If two jobs with the same name are executed in succession, output from the second job will overlay that produced by the first. In such cases, the user should purge the output from the first job after it has been transmitted to him so that a request for output from the second job will not simply return a second copy of the first job's output.

II.C - The Output Connection

RJOR may open the output connection either one or two times as the result of a single transmission on the Input Connection. In each case, the first byte transmitted will consist of zeros indicating message Type 0, and thereafter for the life of the connection, IMP-message boundaries are not significant. Following the first byte, RJOR will transmit the name of the job to which the response applies. The job name will be contained in an 8-byte field identical to that supplied by the user over the Input Connection. Following the job name, RJOR will transmit zero or more variable length logical records. Each will consist of a one-byte op code, a two-byte length field which specifies the unsigned length in bits of the variable length text field which follows. The text field may be zero or more bytes in length; the length field will contain an integer which is a multiple of eight.

The op codes presently defined are listed in Figure 1. An op code of x'01' indicates that the text field contains one record of one of the SYSOUT data sets created by the job whose output was requested. The length fields of all logical records with an op code of x'01' will be identical. For data sets with record lengths other than this value, records are padded on the right with blanks or truncated on the right to this standard record length. Carriage control characters which would ordinarily appear in column 1 for printer-destined output have been discarded and do not appear.* The

*This restriction is temporary; a fix is in the works and will be announced.

records are transmitted to the user in the same sequence as they would be printed on the printer, and collectively include everything that would appear in printed output with the exception of HASP separator sheets.

In all logical records but those with an op code of x'01', the length field contains the value zero, and the op code conveys the entire meaning of the logical record.

Figure 1. Output Connection Op Codes

Op Code (Hex)	Name	Explanation
00	End-of-file.	All output from the job has been transmitted (follows last op-code-x'01' logical record).
01	Output.	The text field contains one record of one SYSOUT data set generated by the job.
02	Output file purged.	Output from the job has been purged as requested.
03	No core for buffer.	Insufficient main storage is available for transmitting output from the job. The transmission request has been aborted and the purge request (if any) has been suppressed.
04	I/O Error reading file.	An irrecoverable I/O error was encountered reading the output file. The transmission request has been aborted and the purge request (if any) suppressed.
05	I/O Error purging file.	An irrecoverable I/O error was encountered purging the output file. The purge request was aborted.
06	No queue space for request.	Output from the job was not available and the wait-for-output option was specified, but RJOR's resources were insufficient to queue the request, which was suppressed.
07	Waiting for output.	Output from the job is not available and the wait-for-output option was specified. RJOR is waiting for the job's output.
08	Canceled request not found.	The user requested that a previously made request specifying the wait-for-output option be canceled. No such request was found by RJOR.
09	Request canceled.	At the user's request, a previously made request specifying the wait-for-output option has been canceled.
0A	I/O Error seeking file.	An irrecoverable I/O error was encountered attempting to locate output from the job. The user's request was aborted.
0B	Output not found.	Output from the job was not found. The wait-for-output option was not specified by the user.